DOCUMENT RESUME

ED 153 602 IR 005 780

AUTHOR Kearsley, G. P.; Hunka, S.
TITLE Documentation in CEI.
INSTITUTION Alberta Univ., Edmonton. Div. of Educational Research Services.
REPORT NO DERS-06-047; RIR-77-9
PUB DATE 77
NOTE 22p.; Best copy available

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS Computer Assisted Instruction; *Computer Oriented Programs; *Computer Programs; Course Descriptions; *Documentation; Higher Education; Indexes (Locaters); *Information Dissemination; *Information Needs
IDENTIFIERS Computer Based Instruction

ABSTRACT

In surveying the problems of documenting computer-based instruction, the need for two levels of documentation is outlined: one concise level designed primarily to catalog available courseware; and a second detailed level intended for the instructors, proctors, programmers, and operators. Three general approaches are considered: manual procedures, self-documenting systems, and post-hoc analysis of programs. The importance of documentation to the development of CEI is discussed. (Author/VT)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED153602

DERS 06-047
October 1977

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

BEST COPY AVAILABLE

RIR-77-9
Documentation in CBI

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

E.W. Romaniuk

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM"

G. P. Kearsley & S. Hunka
Division Of Educational Research Services
University of Alberta

IR005780

ABSTRACT

The problems of documenting computer based instruction are surveyed. The need for two levels of documentation is outlined; one level which is concise and designed primarily to catalogue available courseware and a second level which is detailed and intended for the instructors, proctors, programmers, and operators who will work with the courseware. Three general approaches to producing documentation are considered: manual procedures, self-documenting systems, and post-hoc analysis of programs. The importance of documentation to the development of CBI is discussed.

Documentation in CBI

G.P. Kearsley & S. Hunka
Division of Educational Research Services
University of Alberta

In its broadest meaning, documentation refers to the communication of information. We all know how important communication has become in this age of information overload and explosion. This is no less true in the case of computer systems where the information to be communicated pertains to the existence, understanding, and appropriate use of computer software. Computers have become a fundamental component of most large corporations, utilities, and government agencies and hence it is not surprising that considerable emphasis is now being placed on documentation. This emphasis on documentation is very much an economic necessity. Documentation reduces the human costs involved in producing and maintaining computer software. Since the human activity involved in software has become more costly than the computer hardware itself, this cost saving can be substantial.

As computers begin to become fundamental components of our educational system, we can expect to see a similar concern with documentation of computer based instruction (CBI) software. Many of the things which apply to the documentation of computer software in general also apply to CBI programs. However, there are a number of important differences. One difference involves the documentation of user (student) activity in terms of learning outcomes or examination results. Non - instructional software is seldom concerned with keeping records about the user to the extent expected of instructional programs. Sophisticated CBI programs can involve a considerable amount of graphic displays, formatted text, special character sets, and varied response modes, as well as involving the synchronization of computer-controlled audio, slide, or video devices. Finally, instructional strategies and logics for the individualization of instruction require description which extends beyond the that of the usual algorithms or methods embodied in non-instructional programs. For these reasons alone, the documentation of CBI materials poses a number of unique problems for the documentation of computer software.

There are many reasons why documentation is absolutely critical to the field of CBI. The development of CBI courseware is an expensive and time consuming endeavor typically involving 50-100 hours of design and programming time for every hour of instruction delivered. Clearly, this development effort can be justified if courseware can be shared by many students at different institutions. Documentation of courseware is essential to ensure that such transfer of courseware is possible and successful. Documentation is also necessary to prevent the duplication of similar or identical courseware.

Consider for example, the case of an new instructor who is assigned the responsibilities of teaching the required graduate level course in inferential statistics and who decides to investigate the possibility of using CBI. This is obviously a course for which CBI courseware must exist, the instructor reasons, since it is taught in almost all departments with graduate students at all universities. At this point, the instructor needs documentation which catalogues the existence of this specific course. If this instructor is not personally involved in the CBI area, help might be sought from colleagues in the faculty of education, the staff of the computer center, or from the library/learning center staff. Assuming that the instructor finds a source of information on CBI courseware which lists some graduate statistics courses, the next step involves determining the content of these courses, their length, and very importantly, their instructional quality. Along with these pedagogical questions, the instructor will also need to know the programming language and system involved. This leads to the question of what problems might be involved in running the program at another installation. There are three major possibilities:

1. same language and same system
2. same language, different system
3. different language and system

The first case is fairly straight forward (although it should be noted that identical systems are rare and some differences will be encountered). The second case is fairly typical. Differences in systems might involve different cpu's, compilers, or operating systems, e.g., transfering BASIC programs from one manufacturer's system to another. The third case will involve conversion of courseware such that the same instructional sequences are produced via another language. The degree of resemblance between the original and converted courseware can range from being identical screen for screen to only the instructional strategy or logic being preserved. Assuming that the author decides to go ahead and actually obtain courseware from another installation, there will be the problems of copyright releases and the actual physical transfer of the code (usually via magnetic tape). After having successfully loaded the program onto an available system, the instructor will then need detailed documentation of the actual screen displays, the code, operating procedures, and much more.

The point of this somewhat extended example is to introduce the kind of documentation needs which arise in a CBI context. In the case of transfer of courseware the following questions are likely to arise:

1. What courseware is available?
2. Is it operational or under development?
3. How many hours of instructional time?
4. How good is it?
5. What will it involve to get it running here?
6. Who do I contact to obtain the material?

These are the immediate questions documentation should answer.

Not only is documentation necessary to permit the successful transfer of courseware, it is also essential to the maintenance of a CBI project. Given the realities of changing instructors, programmers, and even computer systems, a CBI project depends upon adequate documentation for ongoing continuity and stability. Furthermore, documentation of student activity is necessary for the evaluation of student progress and the instructional effectiveness of the courseware. Thus, if CBI is to fulfill its potential as a sophisticated instructional design methodology, the documentation must indicate exactly how well the courseware works and exactly where improvement is called for.

In so far as documentation is important to the transfer and dissemination of CBI courseware, to the ongoing operation of CBI projects, and to the improvement of the instructional effectiveness of CBI courseware, it seems necessary to conclude that the eventual success or failure of CBI as an educational methodology may depend in no small way on the success of the documentation of CBI programs. This article discusses the major components of the documentation problem: what is to be documented and how this documentation is to be produced.

Levels of Documentation

To begin with, we distinguish two levels of documentation which provide information for different purposes. Level I documentation is general and concise and designed to identify or catalogue CBI courseware. It's main purpose is to provide a potential user with sufficient information to determine whether further inquiries are called for. Thus, Level I documentation is adequate to the extent that it allows an individual to decide whether a program is of interest or useful to them. Obviously it must be reliable and valid; any inaccuracies undermine the entire documentation effort. It is recommended that Level I documentation contain the following information:

1) subject matter of the course
2) description of content (very brief)
3) status of the course (development/operational)
4) author(s) of the course (person responsible for the definition of subject matter and instructional strategies)
5) exact address of author or person to contact for further information
6) availability and conditions for release of courseware
7) characteristics of intended student group (preferably mentioning appropriate pre-entry skills or grade levels)
8) types of instructional strategies / logics used
9) amount of time required to complete course (average and

range)
10) support materials required (student handouts, manuals, textbooks, or special equipment)
11) support media (type of audio, film, video, graphics, etc., which are needed by the program/system)
12) the programming language
13) the system used (cpu and operating system)
14) the type of terminals used
15) applications (extent to which the course has been used for formal instruction)
16) published references (including any evaluation information)
17) date of this documentation

This type of documentation is exemplified by the Index to Computer Based Instruction which is distributed by the Instructional Media Laboratory at the University of Wisconsin-Milwaukee or the documentation of the Minnesota Educational Computing Consortium (see Koch, 1977). Figure 1 shows a sample documentation from the 1973 Index (Lekan, 1973). Level I documentation is brief and relatively inexpensive to produce and distribute. Because of its categorical nature, it can easily be organized into an information retrieval system for on-line access or inquiry.

The second level of documentation is designed to provide detailed information required by people who are actually working with the courseware. This includes five different groups of users:

A. Instructors: those in charge of providing instruction in specific subject matter areas; this may or may not include the actual developer (author) of the instruction
B. Proctors: those individuals who support instructional activities but who are not themselves subject matter experts (e.g., graduate students, teachers aides)
C. Computer Operators: those individuals who are responsible for the operation of the hardware and software of the CBI system (this might also involve system prorammers/analysts/managers)
D. Programmers: those individuals concerned with the instructional programming and probably responsible for making courses from other institutions operational
E. Students: those individuals who are the target of the instruction

A number of important points should be mentioned regarding these different types of users. The distinctions between the first four categories of users will depend upon the size and nature of the CBI project. In a large scale project (e.g., PLATO, TICCIT), its is likely that the instructor (who authors a course) will be different from the person who programs the course and also different from the proctor who is present when the course is used by students. On the other hand, in the case of a school using drill & practice programs (either via a local minicomputer or remoted located large computer), teachers are likely to design, write, and use the programs themselves. Furthermore, in the case of a

COMPUTER OPERATIONS
AND PROGRAMMING 0316

PROGRAM NAME: Teaching Coursewriter II
AUTHOR: Dr. Steven Hunka
SOURCE: University of Alberta, Division of Educational Research, Ninth Floor, Education Building, Edmonton 7, Alberta, Canada; Dr. Steven Hunka, Director
DESCRIPTION: Introduction to COURSEWRITER II and use of dt, dti, ep, epi, answer analysis, epp, switches, counters, branching, functions, and card-input.
PREREQUISITES: None
LEVEL OF INSTRUCTION: College/university, graduate school
TYPE OF STUDENT: Average
TOTAL HOURS OF AVAILABLE MATERIAL: 10 hours
AVERAGE STUDENT COMPLETION TIME: 8 hours
INSTRUCTIONAL STRATEGY: Problem solving, tutorial, simulation
INSTRUCTIONAL LOGIC: Adaptive
USE OF PROGRAM: Instructional program
SUPPLEMENTARY EQUIPMENT/MATERIALS: Lab exercises
STATUS OF PROGRAM: Additions being made, part operational, part developmental
AVAILABILITY OF PROGRAM: By agreement with organization
FUNDING/SPONSORING AGENCY: Own institution
DESCRIPTIVE LITERATURE: None
LANGUAGE PROGRAMMED IN: COURSEWRITER II
CENTRAL PROCESSOR: IBM 1130 (IBM 1500 Instructional System)
TERMINAL DESCRIPTION: IBM 1510 CRT
APPLICATIONS: Was used by high school (10-12); college/university

FIGURE 1. Example of a program listing from the INDEX TO COMPUTER BASED INSTRUCTION (Lekan, 1973).

stand-alone mini- or microcomputer, one person may not only be the instructor, proctor and programmer but also the computer operator too. None-the-less, these five functions are distinctive and necessary components of a CBI delivery system whether they are fulfilled by a single or different individuals and hence documentation for each function is necessary.

A second consideration is that the extent of the documentation required for each of these groups will vary with the level of instruction involved. Thus, short drill and practice programs delivered via teletypes are not going to require the same amount of documentation as a 100 hour course which involves graphics, audiotapes, slides or laboratory equipment. In addition, certain capabilities (e.g., colour, videodiscs, remote terminals) introduce particular documentation problems not present in other systems.

Considerations such as these will affect the exact details to be included in Level II documentation. The following suggestions elaborate upon the information required in the case of a large scale CBI system:

A. Instructor Documentation

1) major educational objectives of the course
2) pre-entry skills required for each objective
3) for each definable instructional unit within the program, a description of:
   i) subject matter content
   ii) instructional strategies used
   iii) media support required (audio, workbooks, etc.)
   iv) average time for completion
4) availability and nature of performance records used to monitor student progress
5) complete screen by screen display of programs
6) availability and source of evaluation results/studies
7) degree of instructor intervention/assistance required
8) disclaimers and restrictions on use and modification of program

B. Proctor Documentation

1) need for and meaning of any student registration procedures and passwords
2) a description, perhaps by flowchart, showing the entry points in the course, the sequence of instructional units, and the variations of sequence possible due to individualization or branching
3) the need for and location of proctor intervention, including a list of messages and appropriate action to be taken
4) a complete list of all questions asked in the program and correct and incorrect answers

5) a description of procedures for allowing a student to bypass or review a unit in the course
6) the location within the course of all examinations
7) procedures for handling cheating or system failures during exams

C. Operator Documentation

1) the nature and location of the course code (source, object, etc.)
2) the operating details of the hardware involved (e.g., cpu, discs, tapes, terminals, printers)
3) the details of the operating system and all system software (including startup, backup, and emergency procedures)
4) parameters affecting the computer resources needed for each program (memory, graphics, character fonts, etc.)
5) recovery procedures for system failures (hardware or software)

D. Programmer Documentation

1) availability of source code listings for each course
2) the names and locations of all functions, macros, variables, labels, etc., within the course
3) the particular programming strategies/logic used to produce certain instructional displays or effects
4) a complete listing of all error messages which might occur, their meaning and corrective action

E. Student Documentation

1) course outline giving organization of course
2) description of how to use system and terminals
3) description of any scheduling arrangements (when the system is available for that course)
4) how to communicate to the author/instructor via the system (if possible)
5) how to use any special system features (e.g., calculation mode, learner control keys, etc.)

It should be clear that some items of information will be needed by more than one category of users -- for example, the screen by screen listing of displays may be needed by the instructor, proctor, operator, programmer, or student at one time or another. On the other hand, certain items will only be of interest to one user category, e.g., operating procedures. In the case of documentation which is needed by more than one category of users, it will be necessary to provide for access to such documentation (or have multiple copies).

The Production of Documentation

The preceding discussion has covered some of the things which should be included in Level I and II documentation. However, there remains the problem of how to produce such documentation. There are three approaches which can be followed: manual documentation, self-documenting author systems, and post-hoc documentation produced from the code. Manual documentation involves developing administrative procedures or assigning task responsibilities to ensure that the different types of information are recorded. Self-documenting author systems require that a description be provided every time a variable is assigned, a loop initiated, a switch or counter used, etc. These descriptions become comments in the actual computer code. It also allows the system to diagnose potential errors while the course is being authored, e.g., non-terminating loops, missing branch locations, unassigned variables. The third approach involves developing programs which produce documentation from the code in terms of mapping out instructional strategies, identifying variables, loops, switches/counters used, etc. Where possible, the latter two approaches are preferred over the first since the latter automatic methods guarantee that a pre-specified adequate level of documentation is obtained. Furthermore, this will be documentation of the program "as is" rather than as it was originally intended.

Level I documentation is probably best done manually by a single individual assigned with the specific responsibility, perhaps the project director or co-ordinator. A standardized form can be used. A problem in producing Level I documentation is when to collect this information and when to update it. One adminstrative procedure is to request this information when the project is initiated; however, updates will be necessary as the course is developed and used. Level I information may be distributed to other major CBI projects, or put on-line so that members of educational networks/consortiums can inspect this information. Another possibility is that a particular government agency or body could act as a central clearing house or center for such information.

Because of the detailed nature of Level II documentation and the importance of completeness and accuracy, it is desirable that as much as possible be done automatically. Programs can be written which produce:

a) complete screen by screen representations of an entire course (including graphics and scripts of audiovisual material)
b) summaries of student progress including individual progress relative to objectives, and group frequencies of student responses for all questions and exams
c) listing of all questions asked and the correct/incorrect answers
d) debugging aids which give complete locations of all variable/switch/counter assignments, loops, labels

branched to, graphics and character fonts used, and audiovisual use.

e) flowchart representation of the instructional strategies in the program as indicated by the conditional branches in the program

f) catalogue of all graphics and audiovisual materials used in the program in terms of keyword reference and description

One general recommendation for Level II documentation is that it be as self-contained as possible. This means that the description of instructional strategies, proctor instructions, etc. be given in the actual listing of the program. Similarly, the first portion of an audio tape, filmstrip, or videodisc should contain the documentation for its content. The obvious advantage of this is that when materials are transferred the documentation is also transferred.

Two other important capabilities which should be built into most Level II documentation are summarization and cross-referencing. Detailed documentation of almost any aspect of a large course could result in hundreds of pages of output. The number of different screen displays could easily be in the thousands. While there will be cases where the complete listing is necessary, in most instances this amount of output will be so voluminous as to defeat the purpose. Thus it is essential that programs provide some sort of condensation or summary of information. Cross-referencing essentially provides an index of all terms or units which occur and their location. This is particularly useful when it is necessary to trace or alter all occurrences of an item (e.g., audio or slide number, variable name, etc.).

While many aspects of Level II documentation can be done automatically, it should be clear that most will need to be done manually. Again, standard forms and administrative procedures which require that such documentation be completed at appropriate times in the development of a program are neccessary. In addition procedures are necessary relating to the availability of this documentation and its release to other users.

A practical consideration in the production of documentation which must be considered is cost. The cost of Level I documentation should be relatively minimal; it should not be affected by the nature of the programs. The cost of Level II documentation, however, will depend upon the size and complexity of the programs. Automatic documentation procedures will generally be less expensive than manual procedures. The question of how much documentation should cost is diffcult. The costs of lack of documentation, i.e., duplicate and wasted effort, poorly debugged programs, frustrated instructors, programmers, and students, is rather hard to estimate in dollars and cents -- but is very real indeed.

Conclusion

The article has discussed some of the problems in the documentation of CBI materials and outlined the types of information which would constitute adequate documentation. Some of these suggestions have implications for the design of CBI systems and languages (e.g., self-documenting languages). We have not discussed things such as how to write instructional courseware which is easier to document (i.e., the use of structured programming technique) or the nature of operating systems and hardware which facilitate documentation. We also have not discussed Level III documentation (meta-documentation) which would include the generation of reports on system use or reliability, reports on evaluation of courseware, description of the instructional design aspects of courseware, and documentation of available documentation. Level III documentation is within the realm of technical or scientific writing.

As has been pointed out, documentation plays a major role in the success of dissemination and transfer of courseware, the development of new CBI projects, the continuing operation of existing CBI projects, and the optimization of courseware. Furthermore, documentation is fundamental to the very nature of CBI. A CBI program can be thought of as a way of precisely documenting a teaching sequence. However, in contrast to traditional teaching, it is reproducible and in a form which permits expert scrutiny. The implication is that instruction and instructional effectiveness can be studied with considerably more success than in the past. Thus, the effects of modifying teaching strategies, logics, media selection, and so on can be isolated and their effects ascertained. Thus, CBI provides a medium for instructional documentation. Not only is documentation essential for the future success of CBI, but CBI (as instructional documentation) may well be essential for the future evolution of educational theory and practice.

ACKNOWLEDGEMENTS

This paper discusses the cumulative and collective work of many members of the Division of Educational Research Services at the University of Alberta. Particular thanks are due to E.W. Romaniuk and T.C. Montgomerie for their input on an earlier draft of this article.

REFERENCES

Hunka, S. & Romaniuk, E.W. Computer assisted instruction. Canadian Datasystems , June 1974, 34-38.

Koch, H.T. Documentation for instructional applications on a timeshare system. Proceedings of the 15th Annual AEDES Conference, Dallas, April 1977.

Lekan, H.A. Index to computer based instruction. Boston, Mass.: Sterling Institute, 1970.

McGinnis, N. CAL Authoring Aids. Proceedings of the Second Canadian Symposium on Instructional Technology, Quebec City, June, 1976.

Romaniuk, E.W. & Montgomerie, T.C. After implementing you CAI course: What's next. Proceedings of the 2nd Canadian Symposium on Instructional Technology, Quebec City, June 1976.

APPENDIX 1

The Division of Educational Research Services (DERS) in the Faculty of Education at the University of Alberta has operated an IBM 1500 instructional system since 1968. The system has 20 CRT terminals each equipped with audio and image projector units. The system software consists of an 1130 operating system, utility programs for processing student records, a graphics subsystem, and an enhanced version of the COURSEWRITER II author language (see Hunka & Romaniuk, 1974). This appendix briefly describes some of the automatic documentation programs which have been developed by DERS.

1. CRT Displays. This program outputs the text portion of every screen display in a course duplicating the exact format of the screen. Hardcopy reproductions of graphics can be obtained via a program which produces a plotted facsimile.
2. Group Performance Records. This program provides the CRT display for each question asked in the course, a frequency table summarizing all student responses, and their actual input according to student numbers. See Figure 2.
3. Individual Performance Records. This program provides a complete tracing of the students progress in terms of the time through each question and the latency to answer. See Figure 3.
4. Answer Keys. This program provides a list of all questions asked in a course and all correct/incorrect answers. See Figure 4.
5. Flowcharter. This program provides a flowchart indicating the instructional strategies of the program. It permits the inspection of all possible instructional paths in the program. See Figure 5.
6. Documenter. This program provides a complete inventory of all variables, labels, switches, counters, graphic calls, etc., used in a program segment. See Figure 6.

In addition to these documentation programs, standardized routines for terminal introductions, course evaluations, etc., have been developed which ensure that functions common to most courses are described accurately and do not involve redundant programming. The evolution of the student record system is described by Romaniuk & Montgomerie (1976) and some of the system enhancements to support authors by McGinnis (1976).

EP IDENTIFICATION = conc05-039 PLOT = YES

CRT DISPLAY : 1 2
3 4
5 6

What is the rank of this matrix?

a 3 2 1 3 6 e 2 3

MAID	DESCRIPTION OF STUDENT RESPONSE
ot	
c0	a 3 2
w1	1 3 6
w2	e 2 3
w3	

ATTEMPT:	1	2	3	4	>4	SUM
	0	0	0	0	0	0
	28	15	1	0	0	44
	0	0	0	0	0	0
	13	0	0	0	0	13
	0	0	0	0	0	0
TOTALS:	41	15	1	0	0	57

LIGHTPEN (ROW-COL) AT MAID 'c0'

p386(1)	1.	14 ,	2
p387(1)	2.	14 ,	2
p388(2)	3.	14 ,	2
p389(2)	4.	14 ,	2
p390(1)	5.	14 ,	2
p391(2)	6.	14 ,	2
p392(2)	7.	14 ,	2
p393(1)	8.	14 ,	1
p395(1)	9.	14 ,	2
p396(1)	10.	14 ,	2
p397(1)	11.	14 ,	2
p398(1)	12.	14 ,	2
p399(2)	13.	14 ,	2
p400(2)	14.	14 ,	2
p400(3)	15.	14 ,	2
p401(1)	16.	14 ,	2
p402(1)	17.	14 ,	2
p403(1)	18.	14 ,	1
p404(1)	19.	14 ,	2
p415(2)	20.	14 ,	2
p416(1)	21.	14 ,	2
p499(1)	22.	14 ,	2
p500(1)	23.	14 ,	2
p501(2)	24.	14 ,	2
p502(1)	25.	14 ,	2
p503(1)	26.	14 ,	2
p505(2)	27.	14 ,	2

FIGURE 2. Group performance Records,

```
rule -  0      STUDENT ID:a639   04/29/77
11:16                                0033
11:16    0708   -028      a2       0038
11:16    0708   -028      c?       0018
11:17    0708   -072      a2       0014
11:17    0708   -197      a1       0082
11:18    0708   -197      a2       0524
11:18    0708   -028      a2       0006
11:18    0708   -028      a3       0006
11:18    0708   -072      c1       0016
11:18    0708   -078               0023
11:19    0708   -197      a1       0046
11:19    0708   -197      a2       0041
11:19    0708   -028      a2       0005
11:19    0708   -028      a3       0010
11:19    0708   -072      c1       0010
11:19    0708   -078               0005
11:19    0708   -197      a1       0028
11:19    0708   -197      a2       0028
11:19    0708   -028      c1       0019
11:20    0708   -072      c1       0024
11:20    0708   -078               0016
11:21    0708   -197      c0       0870
11:21    0709   -028      ot       0150
11:22    0709   -028      ot       0150
11:22    0709   -028      ot       0150
11:22    0709   -028      c1       0017
11:22    0709   -074      c1       0058
11:22    0709   -080               0019
11:23    0709   -121      c1       0030
11:23    0709   -127               0018
11:23    0709   -161      c0       0149
11:24    0709   -316      a1       0092
11:24    0709   -316      a2       0094
11:24    0709   -335      c0       0070
11:24    0709   -357      c0       0064
11:24    0709   -028      c1       0029
11:24    0709   -074      c1       0011
11:24    0709   -080               0014
11:24    0709   -121      c1       0014
11:24    0709   -127               0012
11:24    0709   -161      c0       0024
11:25    0709   -316      c0       0050
11:25    0709   -335      c0       0036
11:25    0709   -357      c0       0036
11:25    0710   -028      c1       0124
11:25    0710   -074      c1       0020
11:25    0710   -080               0011
11:26    0710   -384      a1       0070
11:26    0710   -384      a2       0013
11:26    0710   -405      c0       0037
11:26    0710   -426      c0       0064
11:26    0710   -448      a1       0072
```

FIGURE 3. Individual Performance Records

```
zz          (CONT.)
          ot


c07ck -009
          ot
          c0      ANSWER WAS:
                  (REVERSED OR BACKWARDS OR (WRONG AND SIDE ))
          w1
          w2
          w3


c07ck -036
          ot
          c0      ANSWER WAS:
                  ((ARROW AND BACKWARD )OR (ARROW AND BACKWARDS )OR (ARROW
                  AND REVERSED )OR (ARROW AND WRONG AND WAY )OR (ARROW AND
                  WRONG AND DIRECTION ))
          w1
          w2
          w3


c07ck -054
          ot


c07ck -062
          ot
          c0      ANSWER WAS:
                  (INVALID OR INCORRECT OR VARIABLE OR (VARIABLE AND NUMBER
                  ))
          w1
          w2


c07ck -089
          ot
          c0      33
          w1      2
          w2
          w3


try1  -003
          ot
          c0      t;2|'
          w1      t;|2'
          w2
          w3
```

FIGURE 4. Answer Keys

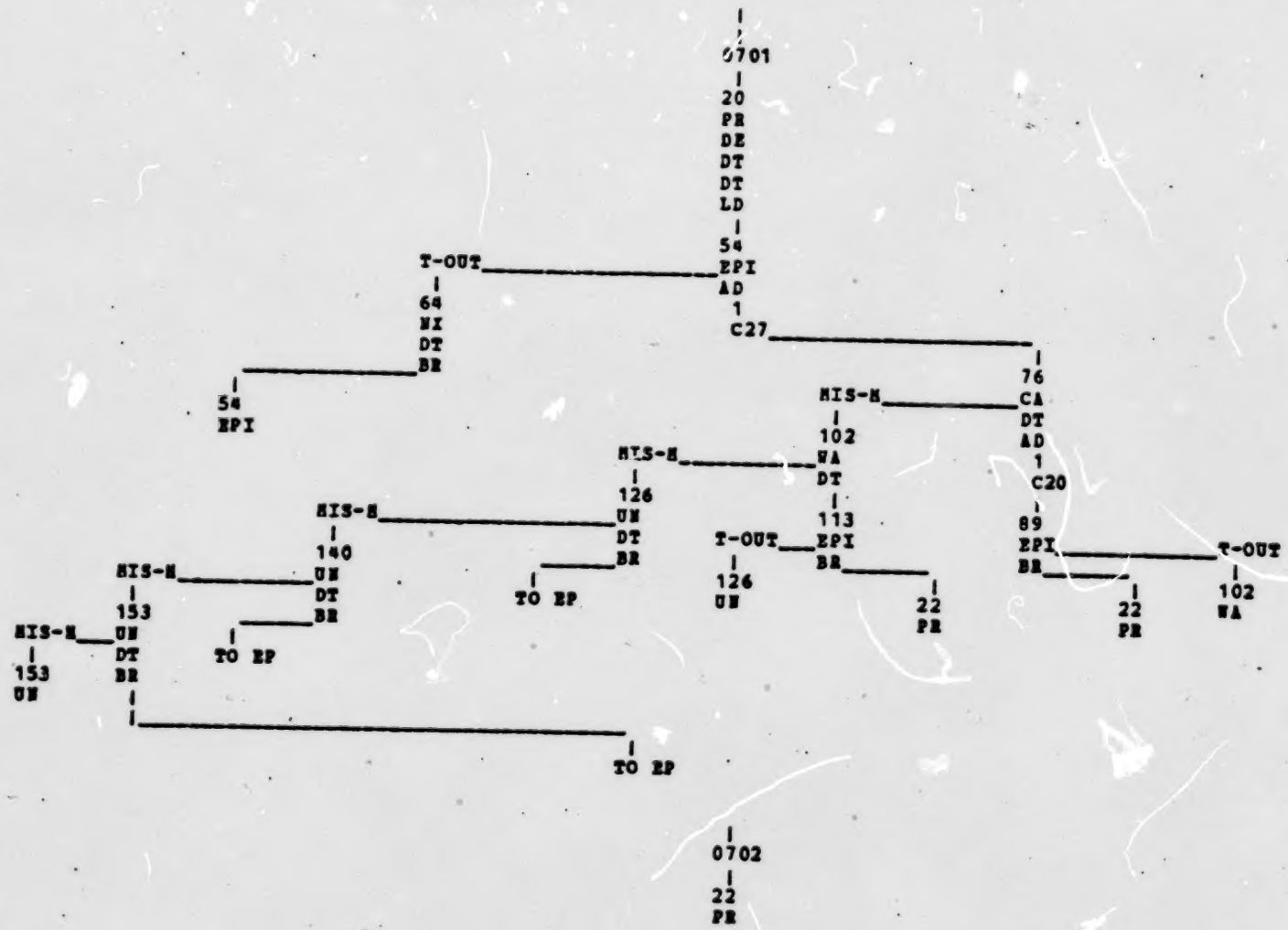

FIGURE 5. Flowchart of Instructional Logic

```
COUNTERS BEING USED AS SWITCHES


RETURN REGISTERS USED

     0   1   2   3   4   5
     0   0   0   0   0   0


FUNCTIONS USED
    FUNCTION        FREQUENCY

        PSRE              17
        UA1-              16
        AV-/              26
        ED-/               3
        XP-/               2
        PITS               3
        UA2-               1
        BA-/               1
        AC-/               4
        ES-/               4


MACROS USED    5

     DRWADP
     DRWASA
     DRWBBA
     DRWOZZ
     DRW2LP


   NUMBER OF PA OP CODES =     116
   NUMBER OF CARDS READ IS    1266


THERE ARE   2 PROCTOR MESSAGES MADE
```

FIGURE 6b. Course Documentation